IE 498 - HW1

> Julius Olson

Implementation

Instructions & requirements

```
# Train
python3 nn.py train


# Load and evaluate
python3 nn.py load



# More advanced usage to customize settings

usage: nn.py [-h] [--hidden HIDDEN] [--epochs EPOCHS] [--data DATA]
             [--model MODEL]
             {train,load}

Neural Network

positional arguments:
  {train,load}     Run NN training or load exisiting model

optional arguments:
  -h, --help       show this help message and exit
  --hidden HIDDEN  Number of hidden layers
  --epochs EPOCHS  Number of epochs
  --data DATA      Dataset destination
  --model MODEL    .npy model destination
```

Requires python ver `3.6` or higher (due to usage of f-string formatting) and numpy.

Implementation

The main part of the program is the class `NeuralNetwork` , which contains all relevant methods needed for training and evaluating the network. Other than that, auxillary functions include different activation functions and a function for loading the data set from file.

Both the forward and backward algorithm utilize vectorization via numpy for faster calculations. The `train` method of the class takes the number of epochs as input, and then performs stochastic gradient descent to update the parameters. At the start of each epoch, an array of indices is created at random. These indices are then used to choose the random points used in the forward and backward steps. The learning rate is also, potentially, updated at the start of each epoch. After each epoch, the training accuracy is displayed and after the training stops the test accuracy is calculated as :

```
# Correct Classified / Total # test samples
```

Result

An accuracy of 97.4% was reached using the following settings.

Param	Value
`d_hidden`	100
`epochs`	10
`LR`	Piecewise constant (see code)
Activation	`sigmoid`